Generative AI

Zant, Colin Martijn van der

CHAPTER 2

An artificial generator and a human as sorting machine

This chapter focuses on generative processes and an unstable sorting machine: a hominid/human. Hominids are renounced for the fluctuations in their categorization capabilities. But is this a problem? Obviously not, since they still seem able to procreate and they do that rather, if not too, successfully. Hominids are still, at the dawn of the third millennium, the most intelligent machines around.

The article chosen for this chapter uses the computer to generative alternatives to an image by means of evolutionary computations. The problem is to binarize a smudged image. The hominid sorts out what he/she prefers, he/she interacts with the evolutionary algorithm resulting in Interactive Evolutionary Computations (IEC). If the task of creating the appropriate image is performed manually by a human, it can take several hours to several days to solve it. Using a computer as generator (easy for the computer) and an intelligent selection mechanism (easy for the hominid) the task takes less than ten minutes to solve.

It is interesting that the fuzziness of the sorting machine actually had to be 'turned on'. First the hominids got the following fitness criterion: "Make the ink black and the rest white". This makes sense from an image processing perspective, but the hominids did not seem to share this perspective and they were poor selectors. Often the quality of the binarization fluctuated too much to make sense of the binarization process. Then they were told: "Make the ink black, the rest white and choose the image that you think looks the best." This very fuzzy task description triggered the corrects methodology in the human to travel fast through the phase space of possible solutions toward a good solution. Suddenly the hominids solved the tasks less time, which means that their selection machines was correctly tuned to the task at hand.

In evolutionary computations the concept of pareto front is important. It is the front in the multi-dimensional phase space where the solutions generated by the algorithm are. This front moves, not unlike a wave, through the phase space to find better solutions in a cyclic manner. This article describes how the algorithms that steer the pareto front have to be altered. The evolutionary computation itself is unlikely to finish, but the selector can decide that it is good enough and halt the computation. If new data arrives though, the computation has to be turned on and a selector needs to check whether the previous solutions are good enough or adjust the automated generative mechanisms, by means of selection, in order to find a solution that satisfies the new data. The pareto front is in IEC is very different from regular evolutionary computations. When a fitness function is used, one tries to spread the pareto front evenly to sample as much as possible of the phase space. When a human is used the IEC performs substantially better by focusing on certain solutions that the selector chooses. Also, in evolutionary computations with every selection step a solution is chosen, but in IEC, if the selector decides to, no solution can be chosen. This increases the performance of the algorithm and reflects the experience of the selector.

This article might teach biological and artificial machines that using each others strengths might result in an abstract searching mechanism which is much better at finding solutions that any of them trying it separately. Usually in evolutionary computation the selection criterion is a fitness function. The fuzzier the fitness is, the more difficult it is to capture it in an algorithm. Humans are well adapted to think fuzzy, to use heuristics. An interesting aspect of this research is that the evolutionary algorithm had to be adjusted. Humans do not enjoy being a fitness function for too long. Computers (still) seem to have no opinion on matters such as joy and boredom and generated the different images without complains.

The article has been published in Volume 6815 of the Society of Photo-Optical Instrumentation Engineers Conference Series in 2008 under the name “ Interactive evolutionary computing for the binarization of degenerated handwritten images”.

2.1 Introduction

Preprocessing and binarization by an expert In the area of document analysis of forensic documents and of old documents it often happens that a person needs a preprocessed and binarized version of a document for further analysis. In the forensic case an expert is working for *hours* up to *days* to create the final documents that are used in the court. All the actions are recorded so it is always possible to know how the final document is created. The final result is a document where only the ink is black, and the rest is white (foreground versus background). The method proposed in this article is fully traceable. It is always known what actions the computer took to get to the desired result. The figures can be found at the end of this chapter.

> The traceability, the knowledge of what has been done so far, can be seen as creating a history in the machines. It is very unlike the history that, for example, mammals build up, but there is no a priori reason that machine intelligence cannot posses a more complex history than what is described in this article.

No expert around For the automatic verification of writers also a binarized version is needed (Bulacu and Schomaker, 2006; Schomaker and Bulacu, 2004). But the police officer who gets a letter from a possible subject might not want to wait for the expert to clean/preprocess and binarize the image. The document might have blood or coffee stains that are difficult to clean, even for experts, but the urgency of the matter is high. The officer would like an answer as soon as possible, but is probably not trained in the use of professional software such as Photoshop [1] or the Gimp[2].

In the case of historical documents an institute might want to publish a cleaned and binarized version on the web, to ease the reading. They have to deal with issues such as leather that has been rotting and eating the paper, or water damage from a big flood, such as the one that occurred in the Netherlands in 1953.

People are smart, computers are fast The method proposed in this article uses the facts that people are smart and that computers are fast. Many people have no problem recognizing what a letter or word is and what the coffee stain or leather rot is. For

[1] http://www.adobe.com/products/photoshop/family/

[2] http://www.gimp.org/

computers this is still very difficult. The computer on the other hand is very good in processing images and applying filters and operations on them in a swift manner. The method proposed in this article combines those two aspects: The computer generates hypotheses that are applied to the document images and shows them to the user. The user decides whether the computer did something that looks nice or something that is totally unreadable and gives feedback to the computer. The computer learns from this feedback and tries to generate better hypotheses. The method for the generation of hypotheses used in this article is evolutionary programming (EP). More specifically; since the human gives feedback a relatively new field of EP is used which is called "Interactive Evolutionary Computing" (IEC). A good overview of the field can be found in (Takagi, 2001a). The technique used in this article is based on the ideas described in (Sims, 1991) where a series of images are presented to the user and the user chooses the image it prefers the most. As discussed in the next sections some specialized adaptations to the standard EP algorithms have to be made to make use of the smart human, who is a lot better in some judgments than computer checked fitness functions. In the rest of this article the word 'individual' refers to an individual in the population of an evolutionary run, and not an user or human. The focus of this article is on the use of IEC and why IEC can be a useful tool to tackle image processing problems and less on the actual image processing operations.

The combination of hypotheses generated by the computer and the selection by a human creates a very powerful abstract search engine. Both are essential during this process. Humans who need this type of image processing, such as people working in law enforcement, might get new ideas to apply it to more and perhaps different problems, since it is possible to process perhaps a hundred images a day, where it used to be possible to process perhaps a single image. These 'new' ideas of the expert user can be seen as the abstract searching mechanism (computer plus expert user) generating new structures on the next level, in this case new crime detection methods. Also information thought to be lost or that was difficult to read can be retrieved, generating new ideas in the mind of researchers of historical texts.

2.2 *Interactive Evolutionary Computing using image processing operations*

2.2.1 *Preprocessing of images*

The preprocessing of images in a non-automated way is a tedious and time consuming process. Many operations can be used and often there are multiple ways to get to a final image that is acceptable for further usage. For a non-expert in the preprocessing it is a field of research which is so enormous that it is unlikely that somebody would dig into it if he/she rarely needs it. Still such a person would like to clean images.

All operations used in this article are fairly standard image processing procedures. The operations sometimes require a single control parameter. The order of execution is essential and the IEC algorithm takes care of both the order and the control parameter. The operations and the range of parameters are listed below:
BackCorrect (10-40) High pass filter implemented by subtracting a blurred version from the original image. Blurring is done using a square uniform window; its size is determined by the control parameter. *Otsu* A global thresholding algorithm based on histograms.
Simple threshold (1-254) Global thresholding based on a single threshold value.
Mean (1-30) Blur the image using a square averaging kernel of given size.
Blur Blur the image using a 5x5 square kernel filled with ones at the border and zeros inside.
Contour Contour detection by convolution with a 3x3 discrete Mexican hat filter.
Detail Detail enhancement by convolution with a 3x3 discrete filter:

0, -1, 0
-1, 10, -1
0, -1, 0
Edge enhance Edge enhancement by convolution with a 3x3 discrete filter: [-1, -1, -1; -1, 10, -1; -1, -1, -1]
Find edges Edge detection by convolution with a 3x3 discrete filter: [-1, -1, -1; -1, 8, -1; -1, -1, -1]
Sharpen Sharpening by convolution with a 3x3 discrete filter: [-2, -2, -2; -2, 32, -2; -2, -2, -2]
Smooth Edge smoothing by convolution with a 3x3 discrete filter: [1, 1, 1; 1, 5, 1; 1,

1, 1]

The examples used for testing Figure 2.2 shows a detail of leather rot and figure 2.3 the binarized version using IEC. It is obvious that this image can use some cleaning. The end user (a non-expert on the cleaning of document images) could not produce any decent results using the Gimp within an hour unless he would edit pixel-by-pixel. Even then the results were very poor. Figure 2.4 shows a detail of a poorly scanned and highly compressed image of the Dutch VOC archives from the 18^{th} century and figure 2.5 the binarized image. Figure 2.6 shows a piece from the "Kabinet der Koningin", which translates as the "Cabinet of the Queen", figure 2.7 the cleaned version. This archive consists of all the royal decrees and decisions from the late 18^{th} century till the end of the 20^{th}. Figure 2.8 is a document from one of the students that was poorly scanned and 2.9 shows the cleaned version.

2.2.2 *Interactive Evolutionary Computing*

IEC is a relatively new branch of the Evolutionary Computing (EC) community, for an introduction see (De Jong, 2002; Takagi, 2001b). In the simple version of the Evolutionary Algorithm (EA) first a child population of n individuals is created. In tournament selection some individuals are selected for a competition against each other and only one of those individuals can go to the parent generation. When the parent generation contains n individuals it is used to generate, by mutation and crossover, a new child population of n individuals. This process continues until a stopping criterion is reached. The steady-state (SS) version is that the winner is put back in the population and it is allowed to mutate/crossover to generate another individual.

The section above clearly demonstrates the relation between Evolutionary Computing and the paradigm of Generative Artificial Intelligence (GAI). The generative processes use feedback to enhance the solutions. The complete sorting process does not make binary decisions but slowly evolves toward fit solutions. The sorting mechanism is steering the process through the solution space and only in the end when the stopping criterion is reached, the solution is given. This is akin to the tracking of the phylum in the case of, for example, gunpowder. The first solutions for the creation of explosions was not the best, but it was a workable one. Slowly in the course of centuries the optimum was found using the feedback mechanisms of trial and error.

The pareto front is the place in the m-dimensional space where the individuals are cluttered during optimization. Since the algorithms used in EC are search algorithms, much effort is being put in keeping this pareto front spread in space. For examples, if two (or more) individuals are close in the m-dimensional space they can get a lower fitness and have less of a chance to procreate. One of the mistakes beginners in EC often make is to have too much of an evolutionary pressure. This means that good solutions are not found because not enough of the space is being sampled and the individuals in the population get stuck in a local minimum.

The pareto front metaphor, and the tools acquired in evolutionary computation, might be interesting for GAI. Instead of having a pareto front with solutions, GAI might be more interested in having a pareto front of possibilities. This front could show possible routes to the next bifurcation points, which could create some form of prediction or more deliberate steering mechanisms.

The interesting part of IEC is that the human is the fitness function (Sims, 1991). The reason is that the fitness function is not known or impossible to define. This is exactly the case with people who are not an expert in image processing but still would like to clean and binarize their document images.

In the algorithms used (IEC+SS and IEC+EA), instead of ensuring that the pareto front is well spread in the space of possibilities, the pareto front is focused. The reason is that the human is so good at deciding on the quality of the processed document image that it seems a waste of time and effort to spread the pareto front.

This change in the way the pareto front is structured through the interactive processes might be a first step toward a different perspective on AI algorithms. Probably in GAI many details compared to Classical AI (CLAI) have to change at least a bit to be useful for GAI.

Focusing the individuals The pareto front is focused by using tournament selection. Normally in tournament selection x individuals are chosen from the population and there is always one individual that goes to the parent generation (EA) or is put back in the current population (SS). The changes made to this in the version of IEC that is discussed in this article is that there is another option: "All the individuals that are shown are bad". This is an essential difference. The user is steering the pareto front to contain only decent or good solutions.

It seems that the better the sorting mechanism is (a human in this case) the more the pareto front can focus. In this case the human knows what the end product should look like, but not how to get there. Could it be that with GAI, which is being trained or steered by humans, the development of intelligent machines also proceeds much faster than what is expected from contemporary AI?

For the initial population random individuals are generated, but the end user decides whether one (or none) of these individuals is used at all for the run of the IEC algorithm. Normally all the individuals are randomly generated without a fitness criterion. Sometimes this random generation of individuals is enough to find a satisfying solution.

The parameters The individuals in the population consist of one or more image processing operations (IPO) in sequential order. The evolutionary operations on the individuals are 1-point crossover, where 2 individuals are cut on a random spot and the last parts are swapped. Another evolutionary operation is a random mutation of an IPO or a control parameter of an IPO. Also two IPO's can be swapped within an individual. All the chances that this can happens are 0.5 for all the experiments. The population size is kept at 30 and the tournament size is fixed at 3. Normally the tournament size is used as a control parameter to adjust the evolutionary pressure. But humans are not so good at evaluating many images against each other. After some experimentation we found that four images was already difficult, so the tournament size is three. With a size of two the experiments were going too slow. Non IEC evolutionary algorithms have a larger freedom in the choice of parameters for the tournament selection.

The end users The end user were initially told to look if the letters were black and the rest was white. This did not produce good results. By accident the author told one of them: “But it also has to look good!”. Since that instruction, using the same system, the users suddenly produced good results in short times instead of poor results during long trials. This is an example of the type of knowledge that is difficult or impossible to model, but very easy for a human to understand.

In GAI terminology: the selection mechanism had to be tuned. This is different from the usual natural setting where the generative theories were postulated first and the selection mechanism is given ”as is”, as for example with the evolution of natural species. The selection mechanism there is the environment. When human started to create settlements they also started to change the selection mechanisms, which resulted in better ways to grow food and cattle with more products which humans needed (wool, milk, meat, fat, . . .).

The setup of the experiments The experiments were setup as follows: the users see an interface as shown in figure 2.10. In the top-left corner the original image is shown, to the right is a big button for the case that all processed images are ugly or bad. The green bar indicated that the user can make a choice (red means ’wait a little longer’). On the bottom three outcomes of different individuals are shown. If the user likes one of the outcomes he/she can click on that image. Only none or one outcome can be chosen. When the user is satisfied he/she can save the result with a menu.

2.3 Results

The results are both conclusive and inconclusive. The conclusive part is that the system works. Non-experts in image processing create good end products using the IEC system on very difficult images. The inconclusive part is that the IEC+SS might outperform the IEC+EA. More data has to be collected to confirm or disprove this hypothesis.

The example in figures 2.3,2.5,2.7, and 2.9 show some typical results of the end products created using the methods of IEC+EA and IEC+SS.

The most important result is that non-experts create these document in minutes instead of hours. Most of the final products, thus when the end user was satisfied, were reached within five minutes. It only happened once during 50 runs that the user was not able to find a decent solution within ten minutes.

The quality of individuals At the start of the algorithm the quality of the individuals is rather poor. During the evolutionary run the quality rises until the user is satisfied. In figure 2.1 it is clear that in the beginning of the run it happened many times that none of the three shown individuals were decent. From selection number 36 and on the user always see a decent individual.

2.4 Discussion

The results in this paper show that IEC is a viable solution for the preprocessing of document images by non-experts in the field of image processing. The computer generates operations on images and the user decides whether the result is decent or not. There did not seem to be a big difference in the performance of different types of evolutionary algorithms. Future research could include the use of IEC on other problems where it is difficult to create an evaluation algorithm, such as sound processing or on the taste of a person regarding art, music and other likes.

> In this research several elements have been demonstrated to already exist: generators in the form of an evolutionary algorithm, an adaptive selection mechanism in the form of humans and a part of the abstract searching machine in the form of generated structures. There is a possibility that the total system of binarization could function as a generator itself in the form of new detection methods used by law enforcement or people interested in degenerated historical images.

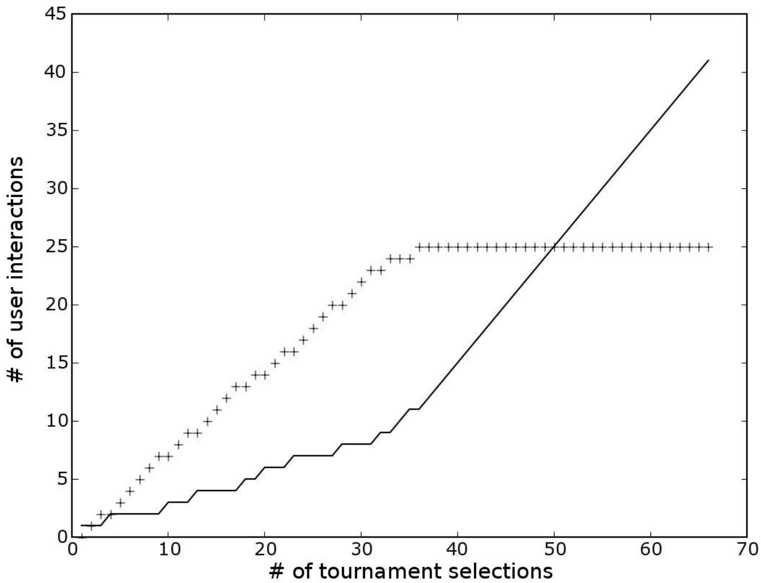


Figure 2.1: The amount of user interactions during a typical run. The connected line is when an individual was chosen (so the user liked what he/she saw) and the + line is when all individuals were bad/ugly according to the user (the user did not like the image). At every user interaction three generated images are shown to the user. This figure shows that the after 36 selections at least one image out of three was good enough and can be regarded as an indication of the improvement of the individuals in the population.

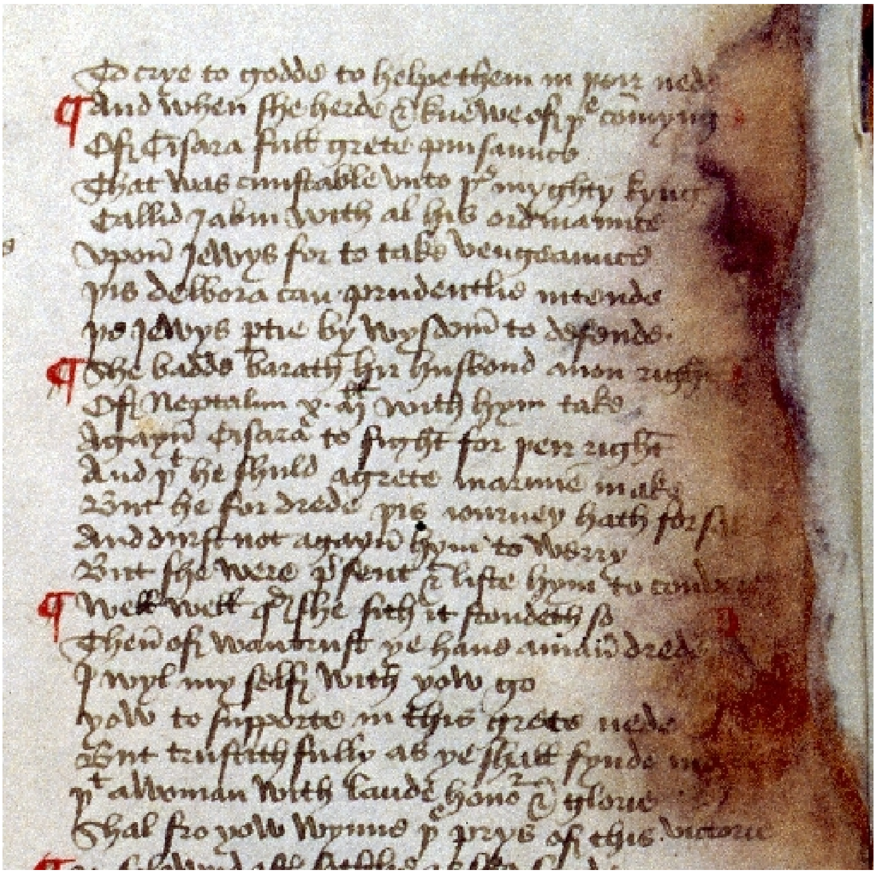


Figure 2.2: Detail of a document from the middle ages before cleaning

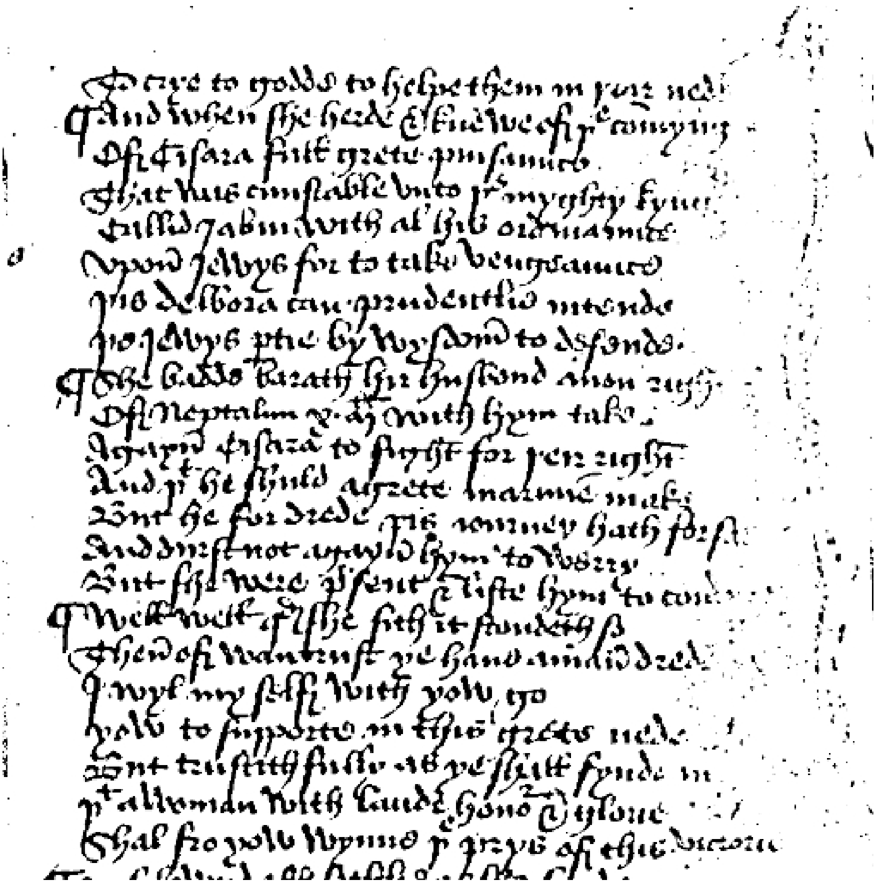


Figure 2.3: Detail of a document from the middle ages after cleaning and binarization using IEC+EA for 301 seconds. The operations used are: `BackCorrect(14), smooth(), smooth()`

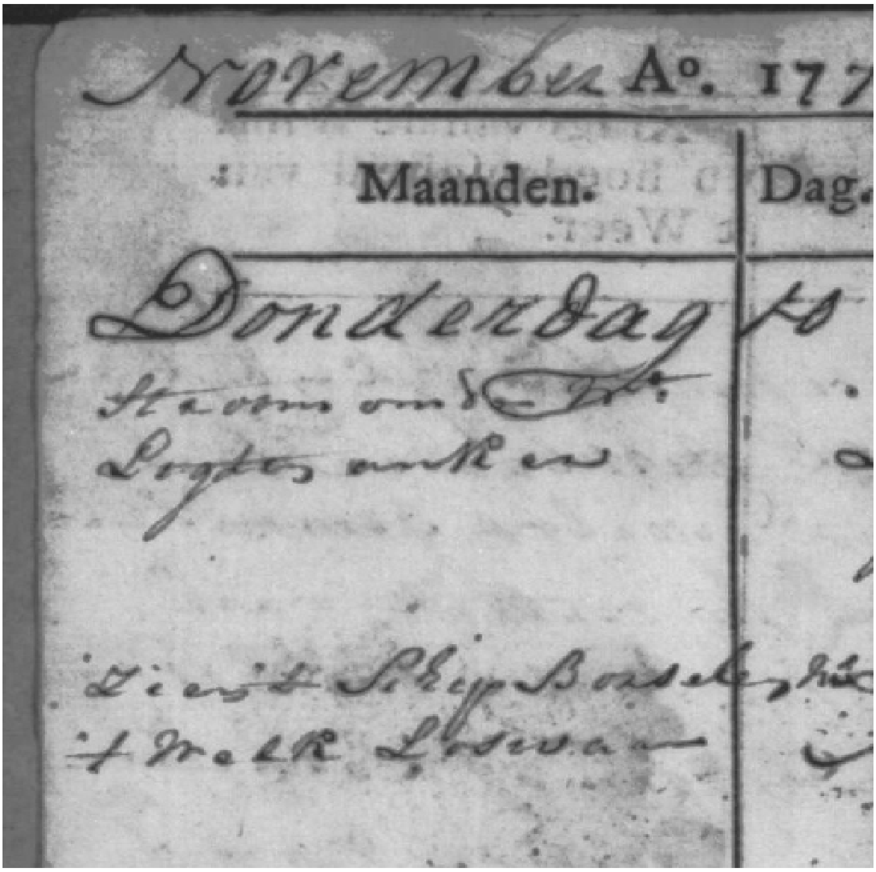

November A°. 177
Maanden. Dag.
Donderdag

Figure 2.4: Detail of a document from the VOC archives (18^{th} century) before cleaning

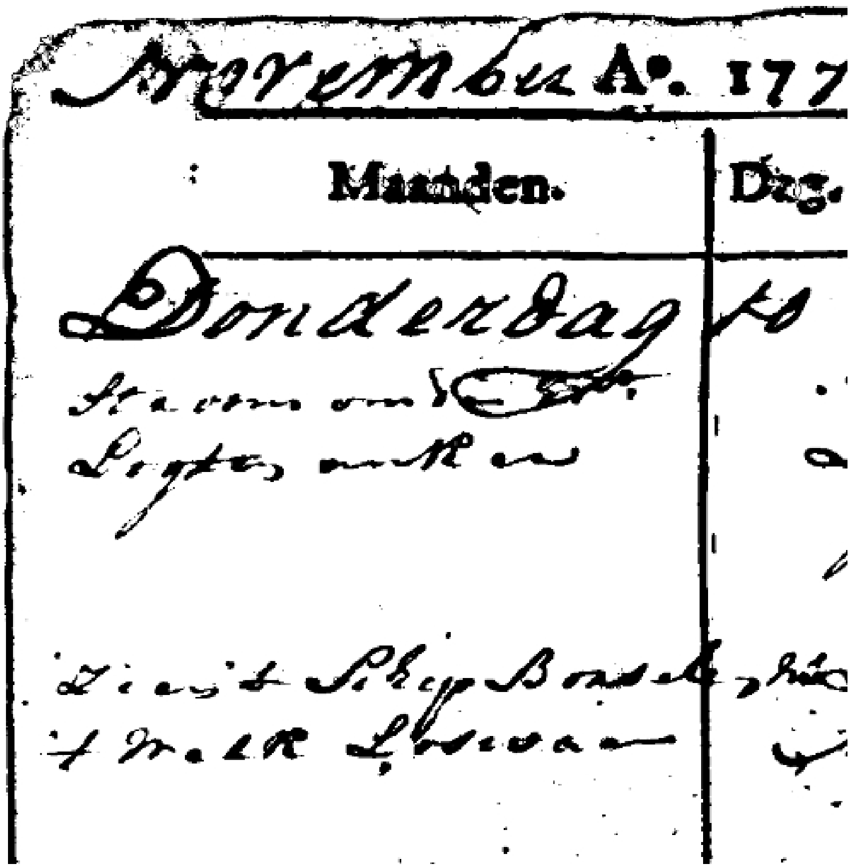


Figure 2.5: Detail of a document from the VOC archives (18^{th} century) after cleaning and binarization using IEC+EA for 295 seconds. The operations used are: `smooth()`, `detail()`, `simpleThreshold(124)`, `BackCorrect(36)`, `edgeEnhance()`

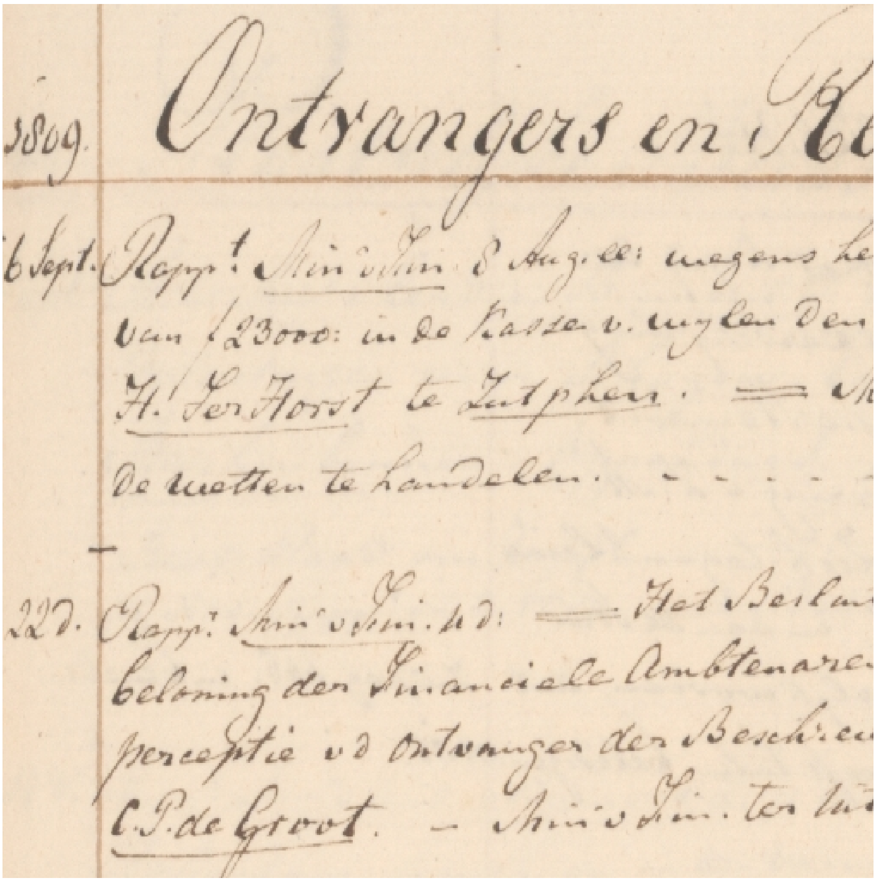


Figure 2.6: Detail of a document from the Cabinet of the Queen from the National Archive (early 19^{th} century) before cleaning

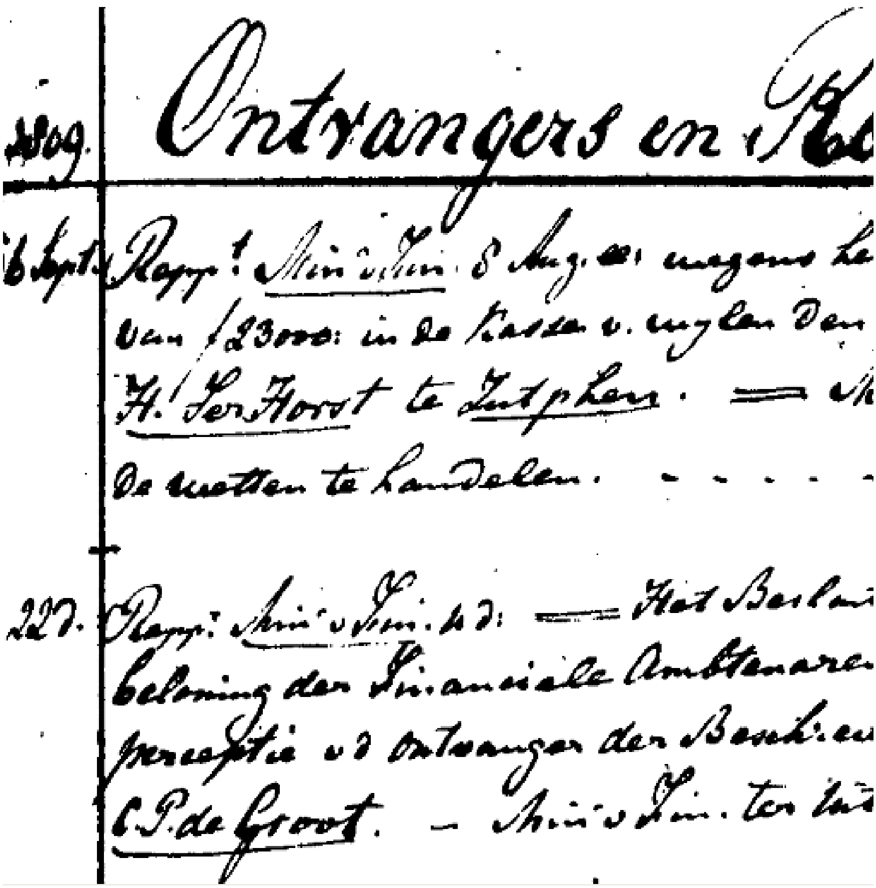
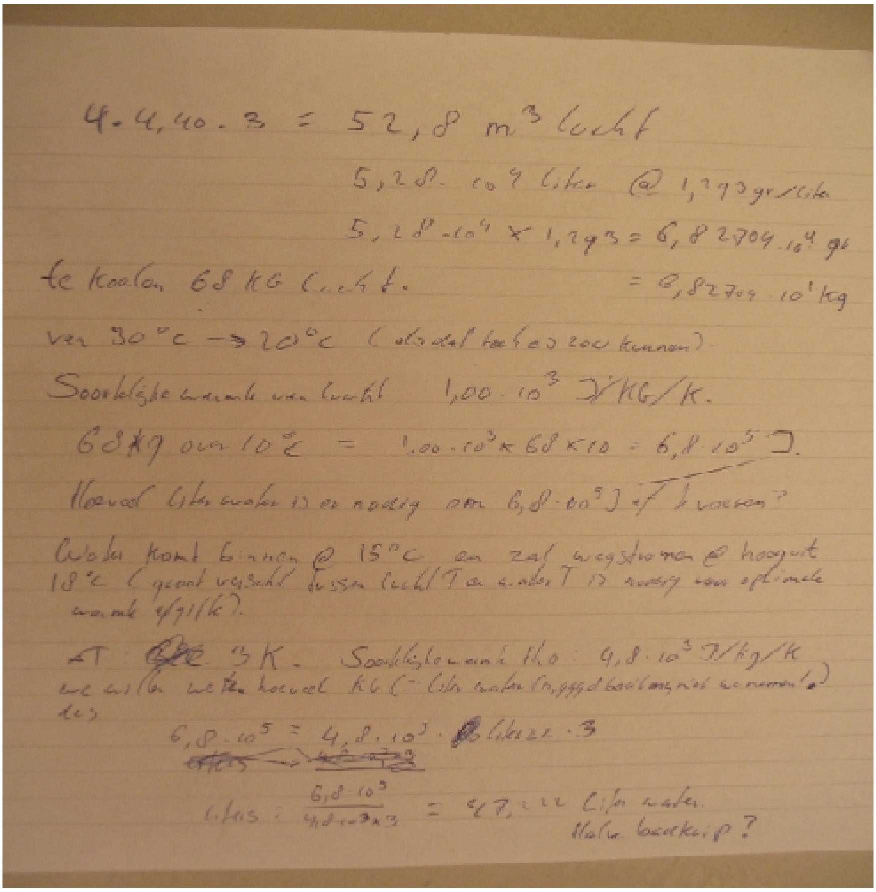


Figure 2.7: An example of a detail of a handwritten document from the Cabinet of the Queen from the National Archive (early 19^{th} century) after cleaning using IEC+SS for 252 seconds. The operations used are: `simpleThreshold(214)`, `smooth()`



$4 \cdot 4{,}40 \cdot 3 = 52{,}8\ m^3$ lucht

$5{,}28 \cdot 10^4$ liter @ 1,293 gr/liter

$5{,}28 \cdot 10^4 \times 1{,}293 = 6{,}82704 \cdot 10^4$ gr

te koelen 68 KG lucht. $= 6{,}82704 \cdot 10^1$ kg

van 30°C → 20°C (als dat toch zo zou kunnen).

Soortelijke warmte van lucht $1{,}00 \cdot 10^3$ J/KG/K.

68 kg om 10°C $= 1{,}00 \cdot 10^3 \times 68 \times 10 = 6{,}8 \cdot 10^5$ J.

Hoeveel liter water is er nodig om $6{,}8 \cdot 10^5$ J af te voeren?

Water komt binnen @ 15°C en zal wegstromen @ hoogst 18°C (groot verschil tussen lucht T en water T is nodig voor optimale warmte afgifte).

ΔT: 3K. Soortelijke warmte H_2O: $4{,}8 \cdot 10^3$ J/kg/K

we willen weten hoeveel KG (~ liter water [illegible])

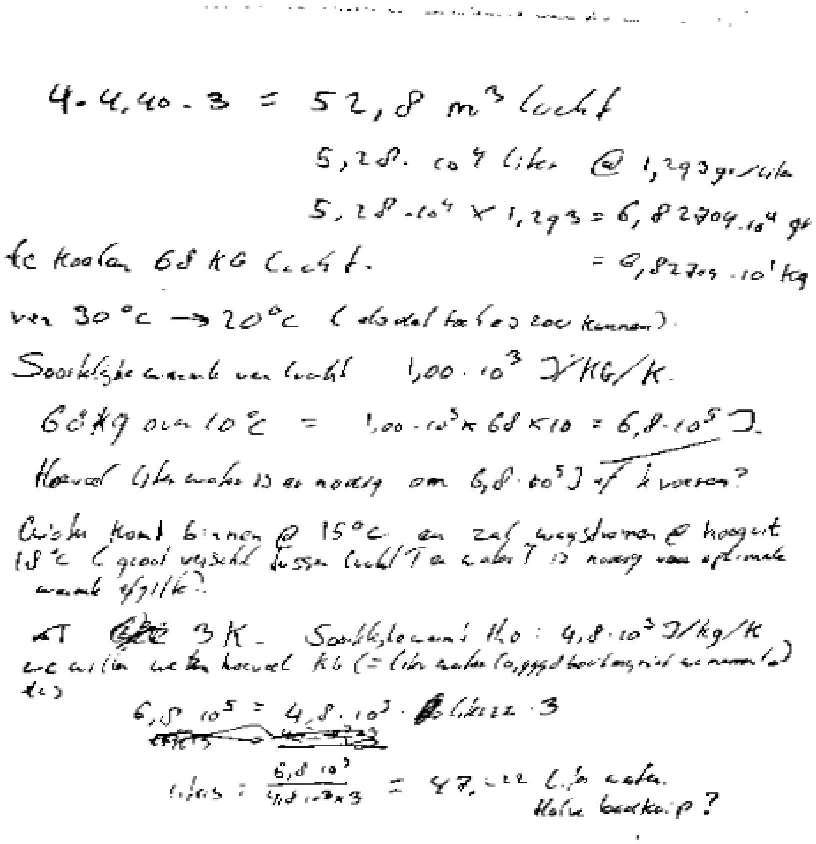
dus

$6{,}8 \cdot 10^5 = 4{,}8 \cdot 10^3 \cdot \text{liters} \cdot 3$

liters $= \frac{6{,}8 \cdot 10^5}{4{,}8 \cdot 10^3 \times 3} = 47{,}222$ liter water.

Halve badkuip?

Figure 2.8: An example of a handwritten document from a student before cleaning


4·4,40·3 = 52,8 m³ lucht

5,28·10⁴ liter @ 1,293 gr/liter

5,28·10⁴ × 1,293 = 6,82704·10⁴ gr

te koelen 68 KG lucht. = 6,82704·10¹ kg

van 30°C → 20°C (als dat tenminste zou kunnen).

Soortelijke warmte van lucht 1,00·10³ J/KG/K.

68 kg om 10°C = 1,00·10³ × 68 × 10 = 6,8·10⁵ J.

Hoeveel liter water is er nodig om 6,8·10⁵ J af te voeren?

Water komt binnen @ 15°C en zal wegstromen @ hoogut 18°C (groot verschil tussen lucht T en water T is nodig voor optimale warmte afvoer?)

ΔT = 3 K. Soortelijke warmte H₂O: 4,8·10³ J/kg/K

we willen weten hoeveel kg (= liter water [illegible])

6,8·10⁵ = 4,8·10³ · liters · 3

liters = 6,8·10⁵ / (4,8·10³ × 3) = 47,222 liter water.

Halve badkuip?

Figure 2.9: An example of a handwritten document from a student after cleaning using IEC+SS for 182 seconds. The operation used is: `BackCorrect(31)`

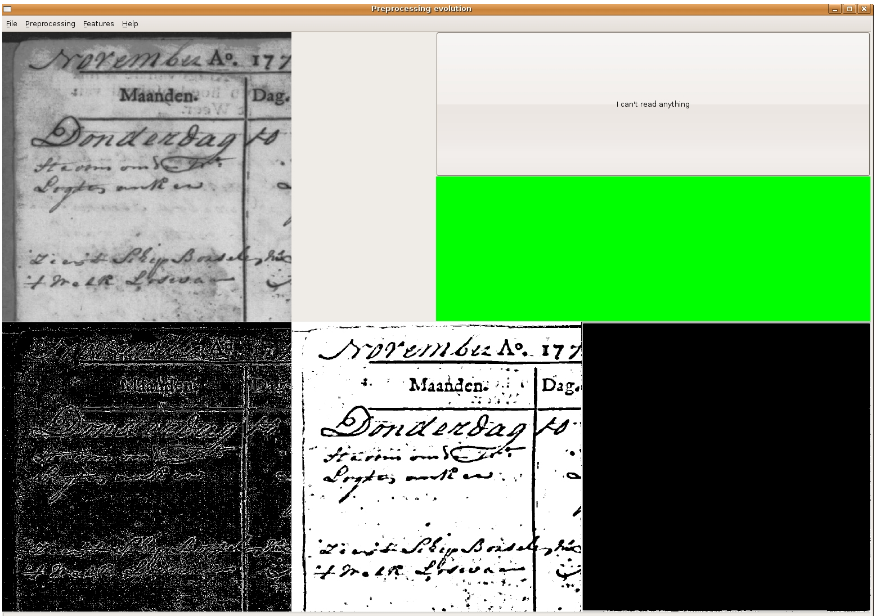


Figure 2.10: The interface during the experiments